

Speedify CLI

The Speedify CLI is a cross-platform utility to help command and monitor Speedify without using the traditional user interface. It is meant to be run on the same device that it is controlling.

The output from Speedify CLI is always one of the following:

- On success, its exit value is 0 and it prints JSON to the console. If there are more than one JSON objects they will be separated by double newlines.
- On error, its exit value is one of the following:
 - 1 = Error from the Speedify API, outputs a JSON error on stderr
 - 2 = Invalid Parameter, outputs a text error message on stderr
 - 3 = Missing Parameter, outputs a text error message on stderr
 - 4 = Unknown Parameter, outputs the full usage message on stderr

For errors from the Speedify API (1), a JSON error message is emitted on stderr. This error contains the fields:

- “errorCode” - a numeric code representing this error,
- “errorType” - a short text code of the error category
- “errorMessage” - a plain text message about the problem

Example Speedify API error message:

```
{
  "errorCode":    3841,
  "errorType":    "Timeout waiting for result",
  "errorMessage": "Timeout"
}
```

On Windows, the executable command is `speedify_cli.exe`. On Linux and Mac, it is `speedify_cli`. For the example commands below, we will use the Linux/Mac name. Add `.exe` if you are using Windows.

On Windows and Linux, please ensure that the background service is running.

On macOS, please ensure that Speedify is open and running to use the CLI.

Usage

The CLI contains the following commands:

```
speedify_cli v16.5.0.154
Usage : speedify_cli [-s] [-t] action
  -s : single line output
  -t : give all outputs a title
Actions:
activationcode
adapter datalimit daily <adapter id> <data usage in bytes|unlimited>
adapter datalimit dailyboost <adapter id> <additional bytes>
adapter datalimit monthly <adapter id> <data usage in bytes|unlimited> <day of the
  month to reset on|0 for last 30 days>
adapter directionalmode <adapter id> <upload mode (on | backup_off | strict_off)>
  <download mode (on | backup_off | strict_off)>
adapter encryption <adapter id> <on|off>
adapter expose-dscp <adapter id> <on|off>
adapter overlimitratelimit <adapter id> <speed in bits per second|0 to stop using>
adapter priority <adapter id> <automatic|always|secondary|backup|never>
adapter ratelimit <adapter id> <download speed in bits per second|unlimited> <upload
  speed in bits per second|unlimited>
adapter remove <adapter id>
adapter resetusage <adapter id>
captiveportal check
captiveportal login <on|off> <adapter id>
headercompression <on|off>
```

```

connect [ closest | public | private | p2p | <country> [<city> [<number>]] | last |
  <tag> ]
connectmethod < closest | public | private | p2p | <country> [<city> [<number>]] |
  <tag> >
connectretry <time in seconds>
daemon exit
directory [directory server domain]
dns <ip address> ...
disconnect
devicerole <client|router>
dscp monitor <dscp value 0-63> ...
dscp queues <add|rem|set> ...
dscp queues add [<dscp value 0-63> [priority] <on|off|auto> [replication]
  <on|off|auto> [retransmissions] <0-255>] ...
dscp queues set [<dscp value 0-63> [priority] <on|off|auto> [replication]
  <on|off|auto> [retransmissions] <0-255>] ...
dscp queues rem [dscp value 0-63] ...
encryption <on|off>
fixeddelay <domains|ips|ports|delay in ms>
fixeddelay domains <add|rem|set> <domain> ...
fixeddelay ips <add|rem|set> <ip address> ...
fixeddelay ports <add|rem|set> [port[-portRangeEnd]/proto] ...
gateway [directory gateway uri]
jumbo <on|off>
login [<username> [password]]
login freeaccount
login oauth [access token]
logout
maxredundant <number of conns>
mode <redundant|speed|streaming>
networksharing set alwaysOnDiscovery <on|off>
networksharing availableshares
networksharing connect <peer connect code>
networksharing discovery
networksharing peer <allow|reject|request|unpair> <peer uuid>
networksharing reconnect <peer uuid>
networksharing set <host|client> <on|off>
networksharing set autoPairBehavior <manual|auto_user|auto_user_team>
networksharing set displayname <new name>
networksharing set hostTunnelRateLimit <rate limit in bits per second>
networksharing set pairRequestBehavior <ask|accept|reject>
networksharing set peer <autoreconnect|allowhost|allowclient> <peer uuid> <on|off>
networksharing settings
networksharing startdiscovery
overflow <speed in mbps>
packetaggr <on|off>
pep <on|off>
pep ips <add|rem|set> <ip address> ...
pep ports <add|rem|set> [port[-portRangeEnd]/proto] ...
pep dscp <add|rem|set> <dscp value 0-63>
packetpool <small|default|large>
ports [port/proto] ...
priorityoverflow <speed in mbps>
privacy advancedIspStats <on|off>
privacy apiProtection <on|off>
privacy requestToDisableDoH <on|off>
refresh oauth [access token]
refresh servers
residentialproxy <on|off>
residentialproxy ports <add|rem|set> [port[-portRangeEnd]/proto] ...

```

```

route default <on|off>
log <erase|daemon>
log erase
log daemon <file size> <files per daemon> <total files> <verbose|info|warn|error>
show < servers | settings | privacy | adapters | currentserver | user | directory |
    connectmethod | streamingbypass | disconnect | streaming | speedtest | logsettings
    | dscp | fixeddelay | trafficrules | pep | systemdata >
speedtest [adapter id]
streamtest [adapter id]
startupconnect <on|off>
state
stats [historic [max-seconds] [<wantedGuid> ...]] | [[durationSeconds]
    [networksharing] [current|day|week|month|total|<periodHours>] ... [<tunnel-guid>
    ...]] | [streaming] | [session [current|day|week|month|total|<periodHours>]]
streaming domains <add|rem|set> <domain> ...
streaming ipv4 <add|rem|set> <ip address> ...
streaming ipv6 <add|rem|set> <ip address> ...
streaming ports <add|rem|set> [port[-portRangeEnd]/proto] ...
streamingbypass domains <add|rem|set> [<domain> ...]
streamingbypass ipv4 <add|rem|set> <ip address> ...
streamingbypass ipv6 <add|rem|set> <ip address> ...
streamingbypass ports <add|rem|set> <port[-portRangeEnd]/proto> ...
streamingbypass service <enable|disable|service name> [<on|off>]
subnets [subnet/length] ...
targetconnections <number upload connections> <number download connections>
transport <auto|tcp|tcp-multi|udp|https>
transportretry <time in seconds>
version

```

Commands

activationcode

Obtain activation code to activate a device on my.speedify.come.

```

$ speedify_cli activationcode
{
    "activationCode" : "123456",
    "activationUrl" : "https://my.speedify.com/activate?activationCode=123456"
}

```

adapter datalimit daily <adapter id> <data usage in bytes|unlimited>

The `adapter datalimit daily` limit the data usage for a specific adapter on a daily basis. The usage can be either limited in bytes or unlimited. This will set the `maxDaily` value accordingly. The adapter guid can be found by using the `show adapters` option. Whether the adapter is disabled or rate limited is controlled by the `adapter overlimitratelimit` setting.

```

$ speedify_cli adapter datalimit daily wlo1 0
[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 261657999,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 261657999,
    }
  }
]

```

```

        "usageMonthlyLimit" : 0,
        "usageMonthlyResetDay" : 0
    },
    "description" : "wlo1",
    "directionalSettings" :
    {
        "download" : "on",
        "upload" : "on"
    }
}
...

```

adapter datalimit dailyboost <adapter id> <additional bytes>

Bumps up the daily datalimit for today only on a specific adapter on a daily basis. The adapter guid can be found by using the `show adapters` option.

```

$ speedify_cli adapter datalimit dailyboost wlo1 0
[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 261744892,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 261744892,
      "usageMonthlyLimit" : 0,
      "usageMonthlyResetDay" : 0
    },
    "description" : "wlo1",
    "directionalSettings" :
    {
      "download" : "on",
      "upload" : "on"
    }
  }
}
...

```

adapter datalimit monthly <adapter id> <data usage in bytes|unlimited> <day of the month to reset on|0 for last 30 days>

The `adapter datalimit monthly` sets a monthly data cap that resets on a set date or lasts 30 days. The usage can be either limited in bytes or unlimited. This will set the max and resetDay accordingly. Whether the adapter is disabled or rate limited is controlled by the `adapter overlimitratelimit` setting.

```

$ speedify_cli adapter datalimit monthly wlo1 2000000000 0
[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 261939351,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 261939351,
      "usageMonthlyLimit" : 2000000000,
      "usageMonthlyResetDay" : 0
    }
  }
}

```

```

    },
    "description" : "wlo1",
    "directionalSettings" :
    {
        "download" : "on",
        "upload" : "on"
    }
}
...

```

adapter directionalmode <adapter id> <upload mode (on | backup_off | strict_off)> <download mode (on | backup_off | strict_off)>

The `adapter directionalmode` controls if and how the adapter is used for upload and download, respectively. Valid settings are “on”, “backup_off”, and “strict_off”.

```
$ speedify_cli adapter directionalmode wlo1 on on
```

```

[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 261952818,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 261952818,
      "usageMonthlyLimit" : 0,
      "usageMonthlyResetDay" : 0
    },
    "description" : "wlo1",
    "directionalSettings" :
    {
      "download" : "on",
      "upload" : "on"
    }
  }
]
...

```

adapter encryption <adapter id> <on|off>

Controls encryption on a single adapter. Note that using the `encryption` command will remove all per-adapter encryption settings. Most of the time, you’ll just want to use the `encryption` command that changes all adapters at same time.

```
$ speedify_cli adapter encryption wlo1 off
```

```

{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : true,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {

```

```

    "port" : 8001,
    "protocol" : "tcp"
}

```

...

adapter expose-dscp <adapter id> <on|off>

The adapter `expose-dscp` allows dscp values of internal packets to be exposed to the VPN transport connection.

```
$ speedify_cli adapter expose-dscp wlo1 off
```

```
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : true,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
}

```

...

adapter overlimitratelimit <adapter id> <speed in bits per second|0 to stop using>

When an adapter `datalimit` is hit, this rate limit (in bit per second) is applied to the adapter. Set to 0 to disable the adapter.

```
$ speedify_cli adapter overlimitratelimit wlo1 0
```

```
{
  "adapterID" : "wlo1",
  "connectedNetworkBSSID" : "",
  "connectedNetworkName" : "",
  "dataUsage" :
  {
    "overlimitRatelimit" : 0,
    "usageDaily" : 261952818,
    "usageDailyBoost" : 0,
    "usageDailyLimit" : 0,
    "usageMonthly" : 261952818,
    "usageMonthlyLimit" : 0,
    "usageMonthlyResetDay" : 0
  },
  "description" : "wlo1",
  "directionalSettings" :
  {
    "download" : "on",
    "upload" : "on"
  }
}

```

...

adapter priority <adapter id> <automatic|always|secondary|backup|never>

The `adapter priority` command allows the user to choose which adapter gets one of the following priorities:

Priority	Description
automatic	Let Speedify manage the connection's priority
always	Use whenever connected
secondary	Use less than Always connection- only when Always connections are congested or not working
backup	Only use when other connections are unavailable
never	Adapter is not used

This will set priority as one of the above mentioned options accordingly.

```
$ speedify_cli adapter priority wlo1 always
```

```
[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 261952818,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 261952818,
      "usageMonthlyLimit" : 0,
      "usageMonthlyResetDay" : 0
    },
    "description" : "wlo1",
    "directionalSettings" :
    {
      "download" : "on",
      "upload" : "on"
    }
  }
]
...
```

adapter ratelimit <adapter id> <download speed in bits per second|unlimited> <upload speed in bits per second|unlimited>

The `adapter ratelimit` command allows the user to throttle the adapter's maximum speed, in bits per second.

```
$ speedify_cli adapter ratelimit wlo1 0
```

```
[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 261952818,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 261952818,
      "usageMonthlyLimit" : 0,
      "usageMonthlyResetDay" : 0
    },
    "description" : "wlo1",
    "directionalSettings" :
    {
      "download" : "on",
      "upload" : "on"
    }
  }
]
```

```
{
  "download" : "on",
  "upload" : "on"
...
}
```

adapter remove <adapter id>

The `adapter remove` command removes an adapter from Speedify's cached adapter list. Use this to clean up stale adapters that are no longer present on the system. The adapter must not be currently connected.

```
$ speedify_cli adapter remove eth1234
[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 261952818,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 261952818,
      "usageMonthlyLimit" : 0,
      "usageMonthlyResetDay" : 0
    },
    "description" : "wlo1",
    "directionalSettings" :
    {
      "download" : "on",
      "upload" : "on"
    }
  }
...
]
```

adapter resetusage <adapter id>

The `adapter resetusage` command resets the statistics associated with this adapter. This restarts any daily and monthly data caps.

```
$ speedify_cli adapter resetusage wlo1
[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 0,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 0,
      "usageMonthlyLimit" : 0,
      "usageMonthlyResetDay" : 0
    },
    "description" : "wlo1",
    "directionalSettings" :
    {
      "download" : "on",
      "upload" : "on"
    }
  }
...
]
```

captiveportal check

Checks which interfaces are currently being blocked by a captive portal. Returns an array of adapterIDs which are currently captive.

```
$ speedify_cli captiveportal check
["{D5306735-8BD9-4F89-A1AF-F485CA23208C}"]
```

captiveportal login <on|off> <adapter id>

Starts or stops directing web traffic out the local interface to allow users to login to the captive portal web page. Setting this to “on” and passing in an Adapter ID, will direct and new connections on ports 53,80 and 443 out the specified adapter. If the Speedify user interface is running it will launch a captive portal web browser component. Setting this to “off” (no Adapter ID required in this case), will stop the forwarding of the web traffic, and will allow it to all pass over the VPN tunnel as usual.

```
$ speedify_cli captiveportal login on "{D5306735-8BD9-4F89-A1AF-F485CA23208C}"
{
  "enabled":      true,
  "adapterID":    "{D5306735-8BD9-4F89-A1AF-F485CA23208C}"
}
```

headercompression <on|off>

The headercompression command sets header compression on/off.

```
$ speedify_cli headercompression on
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : true,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
  ...
}
```

connect [closest | public | private | p2p | <country> [<city> [<number>]] | last | <tag>]

The connect command connects to a server based on your connectmethod setting, or a server of your choosing. It prints details of the server it has selected.

The show servers command will give you a detailed list of servers with their countries, cities and number as fields that you can use in this command.

Since 13.4 the connect command persistently stores the arguments so that the next time the daemon connects, it will connect with the same arguments. Previously, you needed to use the connectmethod command to persistently store the setting, but this is no longer required.

It accepts both the “tag” style of server name, such as us-newark-18 or accepts country, city and num as separate arguments, us newark 18.

To connect to the nearest server in a particular country, pass along a two-letter country code drawn from the `speedify_cli show servers` command:

```
$ speedify_cli connect ca
```

To connect to a particular city, pass along a two-letter country code and city, drawn from the `speedify_cli show servers` command:

```
$ speedify_cli connect us atlanta
```

To connect to a specific server, pass along a tag which consists of a two-letter country code, city, and number, separated by hyphens. These can be drawn from the `speedify_cli show servers` command. By default, if Speedify cannot connect to the named server, it will attempt to connect to another server in the same city. However, by adding a `#` at the front of the tag, it will lock it only connect to the specific named server (note that on some operating systems, such as Linux, you may need to use quotes around the argument):

```
$ speedify_cli connect "#us-atlanta-3"
```

Example:

```
$ speedify_cli connect
{
  "city" : "nyc",
  "country" : "us",
  "dataCenter" : "vultr-nyc",
  "dnsIP" :
  [
    "10.202.8.8"
  ],
  "friendlyName" : "United States - New York City #45",
  "isPremium" : false,
  "isPrivate" : false,
  "num" : 45,
  "publicIP" :
  [
    "149.28.8.8"
  ],
  "tag" : "us-nyc-45",
  "torrentAllowed" : false
}
```

`connectmethod < closest | public | private | p2p | <country> [<city> [<number>]] | <tag>`
>

The `connectmethod` command sets the connection method used during autoconnect, without actually connecting. It prints the state immediately after the request to set the connection method is made. The method used can be `private`, `p2p` or `closest`. In order for this setting to be used, you may need to begin an autoconnect after changing the method, by running `speedify_cli connect`.

```
$ speedify_cli connectmethod closest
{
  "city" : "",
  "connectMethod" : "closest",
  "country" : "",
  "num" : 0
}
```

`connectretry <time in seconds>`

The `connectretry` command sets the backoff timeout limit in seconds for repeated errors when logging in or connecting to VPN servers. With each subsequent failure, the wait time increases before attempting to login or connect again. This call sets the upper limit on that backoff timeout.

```

$ speedify_cli connectretry 300
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : true,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
  ...
}

```

daemon exit

Causes the Speedify service to disconnect, and exit. In general, leave this alone.

directory [directory server domain]

Controls the directory server. In general, leave this alone.

dns <ip address> ...

The `dns` command sets the DNS servers to use for domain name resolution.

```

$ speedify_cli dns 8.8.8.8
{
  "advancedIspStats" : true,
  "apiProtection" : true,
  "appAnalytics" : false,
  "dnsAddresses" :
  [
    "8.8.8.8"
  ],
  "requestToDisableDoH" : false,
  "residentialProxyEnabled" : true
}

```

disconnect

The `disconnect` command disconnects from the server. It prints the state immediately after the request to disconnect is made.

```

$ speedify_cli disconnect
{
  "state" : "LOGGED_IN"
}

```

devicerole <client|router>

The `devicerole` command sets the device role to either `client` or `router`.

```
$ speedify_cli devicerole client
{
  "model": "Example Device",
  "role" : "client",
  "uuid": "00000000-0000-0000-0000-000000000000",
  "vendor": "Example Vendor"
}
```

dscp monitor <dscp value 0-63> ...

The `dscp monitor` command sets DSCP values to monitor. Monitoring a DSCP value allows tracking statistics for traffic with that DSCP tag. Values must be between 0 and 63.

```
$ speedify_cli dscp monitor 46
{
  "dscpQueues" :
  [
    {
      "dscp" : 0,
      "priority" : "on",
      "replication" : "auto",
      "retransmissionAttempts" : 2
    }
  ],
  "monitoringTags" :
  [
    46
  ]
}
```

dscp queues <add|rem|set> ...

Configure how dscp queues are handled.

```
$ speedify_cli dscp queues add 0 priority on replication auto retransmissions 2
{
  "dscpQueues" :
  [
    {
      "dscp" : 0,
      "priority" : "on",
      "replication" : "auto",
      "retransmissionAttempts" : 2
    }
  ],
  "monitoringTags" :
  [
    46
  ]
}
```

dscp queues add [<dscp value 0-63>] [**priority**] <on|off|auto> [**replication**] <on|off|auto> [**retransmissions**] <0-255>] ...

Configure how dscp queues are handled.

```
$ speedify_cli dscp queues add 0 priority on replication auto retransmissions 2
```

```

{
  "dscpQueues" :
  [
    {
      "dscp" : 0,
      "priority" : "on",
      "replication" : "auto",
      "retransmissionAttempts" : 2
    }
  ],
  "monitoringTags" :
  [
    46
  ]
}

```

dscp queues set [<dscp value 0-63> [priority] <on|off|auto> [replication] <on|off|auto> [re-transmissions] <0-255>] ...

Configure how dscp queues are handled.

```

$ speedify_cli dscp queues add 0 priority on replication auto retransmissions 2
{
  "dscpQueues" :
  [
    {
      "dscp" : 0,
      "priority" : "on",
      "replication" : "auto",
      "retransmissionAttempts" : 2
    }
  ],
  "monitoringTags" :
  [
    46
  ]
}

```

dscp queues rem [dscp value 0-63] ...

Configure how dscp queues are handled.

```

$ speedify_cli dscp queues add 0 priority on replication auto retransmissions 2
{
  "dscpQueues" :
  [
    {
      "dscp" : 0,
      "priority" : "on",
      "replication" : "auto",
      "retransmissionAttempts" : 2
    }
  ],
  "monitoringTags" :
  [
    46
  ]
}

```

encryption <on|off>

The `encryption` command enables or disables encryption of all tunneled traffic. It prints the connection settings immediately after the change is made. Note that this will clear all per-adapter encryption settings from the `adapter encryption` command.

```
$ speedify_cli encryption off
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
  ...
}
```

fixeddelay <domains|ips|ports|delay in ms>

The `fixeddelay` command specifies a delay in milliseconds for a jitter buffer. The jitter buffer adds a delay to traffic and delivers it with a consistent latency through the VPN tunnel to minimize jitter. The delay is applied in each direction on top of the baseline latency, so the effective round trip time is twice the `fixeddelay` value plus the underlying connection latency. Rules can be defined by ports, IP addresses and domains. The rules specify which traffic should have the delay applied while other traffic is delivered normally. This feature can be helpful with live streaming and other applications which are sensitive to changes in latency, but can handle extra delay. The jitter buffer should not be used for real-time back and forth communication, such as audio / video calls or gaming where extra delay can degrade the experience.

```
$ speedify_cli fixeddelay 100
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
  ...
}
```

fixeddelay domains <add|rem|set> <domain> ...

Configure domains to have the fixed delay jitter buffer applied.

```
$ speedify_cli fixeddelay domains add example.com
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
}
...
```

fixeddelay ips <add|rem|set> <ip address> ...

Configure IP addresses to have the fixed delay jitter buffer applied.

```
$ speedify_cli fixeddelay ips add 8.8.8.8
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
}
...
```

fixeddelay ports <add|rem|set> [port[-portRangeEnd]/proto] ...

Configure ports to have the fixed delay jitter buffer applied.

```
$ speedify_cli fixeddelay ports add 8200/tcp
{
  "trafficRules" :
```

```
[
  {
    "actions" :
    [
      {
        "type" : "fixed_delay"
      }
    ],
    "conditions" :
    [
      {
        "type" : "ports",
        "value" :
        [
          {
            "port" : 9000,
            "portRangeEnd" : 9010,
            "proto" : "udp"
          }
        ]
      }
    ]
  }
]
...
```

gateway [directory gateway uri]

Configures the OAuth gateway url to use.

```
$ speedify_cli gateway https://my.domain.com/oauth/gateway/path
{
  "domain" : "",
  "enableEsni" : false,
  "gatewayUri" : "https://my.domain.com/oauth/gateway/path"
}
```

jumbo <on|off>

The jumbo command allows the TUN adapter to accept larger MTU packets. This will set `jumbo_packets` to either `True` or `False`.

```
$ speedify_cli jumbo on
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
}
...
```

login [<username> [password]]

The `login` command instructs Speedify to connect with the given username and password. It prints the state immediately after the request to login is made. Speedify will then proceed to automatically connect if the login succeeds.

```
$ speedify_cli login user@domain.com password123
{
  "state": "LOGGED_IN"
}
```

login freeaccount

The `login freeaccount` command instructs Speedify to connect to a free account with a set data limit. It prints the following state immediately after the request is made.

```
$ speedify_cli login freeaccount
{
  "state": "LOGGED_IN"
}
```

login oauth [access token]

The `login oauth` logs in with the user represented by encrypted token passed in. It prints the state immediately after the request to login is made. Speedify will then proceed to automatically connect if the login succeeds.

```
$ speedify_cli login oauth {encrypted_token}
{
  "state": "LOGGED_IN"
}
```

logout

The `logout` command disconnects from the server and flushes any user credentials that were stored.

```
$ speedify_cli logout
{
  "state": "LOGGED_OUT"
}
```

maxredundant <number of conns>

The `maxredundant` command controls the maximum number of connections will be use simultaneously in redundant mode.

```
$ speedify_cli maxredundant 3
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
```

```
        "protocol" : "tcp"
    }
    ...
```

mode <redundant|speed|streaming>

The mode command instructs Speedify to optimize for maximum connection speed or redundancy. Valid options are **speed**, **redundant**, and **streaming**.

```
$ speedify_cli mode speed
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
  ...
```

networksharing set alwaysOnDiscovery <on|off>

Controls if Pair and Share peer discovery is always active. This can use more power when on battery.

```
$ speedify_cli networksharing set alwaysOnDiscovery on
{
  "alwaysOnDiscovery" : true,
  "autoPairBehavior" : "auto_user_team",
  "clientEnabled" : true,
  "displayName" : "peer device",
  "hostConnectCode" : "",
  "hostEnabled" : false,
  "pairRequestBehavior" : "accept"
}
```

networksharing availableshares

The networksharing commands control the Pair & Share behavior of this device, allowing it to share cellular connections over a Wi-Fi network to act as virtual network adapters. **availableshares** returns an array of all currently available networksharing peers that were found via discovery, and what is known about them.

```
$ speedify_cli networksharing availableshares
[
  {
    "autoReconnect" : true,
    "displayName" : "peer device",
    "haveAuthToken" : false,
    "peerAsClient" :
    {
```

```

    "allowed" : true,
    "peerStatus" : "disconnected",
    "tunnelStatus" : "inactive",
    "usage" :
    {
        "month" : 0,
        "today" : 0,
        "total" : 0,
        "week" : 0
    }
},
"peerAsHost" :
{
    "allowed" : true,
    "peerStatus" : "unauthenticated",
    "tunnelStatus" : "inactive",
    "usage" :
    {
        "month" : 0,
        "today" : 0,
        "total" : 0,
        "week" : 0
    }
},
"supportsHost" : true,
"uuid" : "F1C1290C-CA03-17B4-9989-A22222A28B74"
}
]

```

networksharing connect <peer connect code>

Connect to a peer, using a connect code (hostConnectCode), if that peer is available on the local network. Connect codes normally come from scanning the peer's QR code. You probably actually want to use `networksharing peer request <uuid>` if you are trying to connect to peer you found in `availableshares`.

```

$ speedify_cli networksharing connect F1C1290C-CA03-17B4-9989-A22222A28B74
{
    "peerStatus" : "authenticated",
    "role" : "host",
    "uuid" : "F1C1290C-CA03-17B4-9989-A22222A28B74"
}

```

networksharing discovery

Shows the current state of Pair & Share discovery.

```

$ speedify_cli networksharing discovery
{
    "discoveryActive" : false
}

```

networksharing peer <allow|reject|request|unpair> <peer uuid>

Controls pairing behavior with the peer with the given uuid. The uuid can be found with `networksharing availableshares`. The options are: * `allow` - accepts a request to pair from this peer * `reject` - refuses a request to pair from this peer * `request` - sends a request to pair to this peer * `unpair` - disconnects and unpairs from this peer. You will need to `request` / `allow` again before using this peer again.

```

$ speedify_cli allow F1C1290C-CA03-17B4-9989-A22222A28B74 on
{
    "displayName" : "peer device",

```

```
"haveAuthToken" : true,
"uuid" : "F1C1290C-CA03-17B4-9989-A22222A28B74"
}
```

networksharing reconnect <peer uuid>

Reconnect to the peer at .

```
$ speedify_cli networksharing reconnect F1C1290C-CA03-17B4-9989-A22222A28B74
{
  "peerStatus" : "authenticated",
  "role" : "host",
  "uuid" : "F1C1290C-CA03-17B4-9989-A22222A28B74"
}
```

networksharing set <host|client> <on|off>

Controls if device acts as a **client** (using peers' shared cellular) or a **host** (allowing peers to use our cellular). Currently only mobile platforms can act as hosts.

```
$ speedify_cli networksharing set client on
{
  "alwaysOnDiscovery" : true,
  "autoPairBehavior" : "auto_user_team",
  "clientEnabled" : true,
  "displayName" : "peer device",
  "hostConnectCode" : "",
  "hostEnabled" : false,
  "hostTunnelRateLimit" : 0,
  "pairRequestBehavior" : "ask",
  "sameUserAutoPair" : true
}
```

networksharing set autoPairBehavior <manual|auto_user|auto_user_team>

Controls if peers of the same user and/or team are automatically paired.

```
$ speedify_cli networksharing set autoPairBehavior auto_user_team
{
  "alwaysOnDiscovery" : true,
  "autoPairBehavior" : "auto_user_team",
  "clientEnabled" : true,
  "displayName" : "peer device",
  "hostConnectCode" : "",
  "hostEnabled" : false,
  "pairRequestBehavior" : "accept"
}
```

networksharing set displayname <new name>

A valid utf-8 display name. Other devices will see this name when they pair/share with this device. Name is for display only, settings and connections are based on the automatically generated uuid.

```
$ speedify_cli networksharing set displayname peer device
{
  "alwaysOnDiscovery" : true,
  "autoPairBehavior" : "auto_user_team",
  "clientEnabled" : true,
  "displayName" : "peer device",
  "hostConnectCode" : "",
  "hostEnabled" : false,
```

```

"hostTunnelRateLimit" : 0,
"pairRequestBehavior" : "ask",
"sameUserAutoPair" : true
}

```

networksharing set hostTunnelRateLimit <rate limit in bits per second>

Sets the host tunnel rate limit in bits per second. Set to 0 for unlimited.

```

$ speedify_cli networksharing set hostTunnelRateLimit 0
{
  "alwaysOnDiscovery" : true,
  "autoPairBehavior" : "auto_user_team",
  "clientEnabled" : true,
  "displayName" : "peer device",
  "hostConnectCode" : "",
  "hostEnabled" : false,
  "hostTunnelRateLimit" : 0,
  "pairRequestBehavior" : "ask",
  "sameUserAutoPair" : true
}

```

networksharing set pairRequestBehavior <ask|accept|reject>

Set the behavior for incoming connection requests; ask, accept, or reject.

```

$ speedify_cli networksharing set client on
{
  "alwaysOnDiscovery" : true,
  "autoPairBehavior" : "auto_user_team",
  "clientEnabled" : true,
  "displayName" : "peer device",
  "hostConnectCode" : "",
  "hostEnabled" : false,
  "hostTunnelRateLimit" : 0,
  "pairRequestBehavior" : "ask",
  "sameUserAutoPair" : true
}

```

networksharing set peer <autoreconnect|allowhost|allowclient> <peer uuid> <on|off>

Turns on and off settings related to an individual peer, identified by uuid. dir

- `autoconnect` controls whether to automatically reconnect to this peer when it is available.
- `allowhost` controls whether to allow peer to act as a host (e.g. will we use its offered cellular connections).
- `allowclient` controls whether to allow peer to use any cellular adapters this computer is sharing as a host.

```

$ speedify_cli networksharing set peer autoreconnect F1C1290C-CA03-17B4-9989-A22222A28B74
on
[
  {
    "autoReconnect" : true,
    "displayName" : "peer device",
    "haveAuthToken" : true,
    "peerAsClient" :
    {
      "allowed" : true,
      "peerStatus" : "unauthenticated",
      "tunnelStatus" : "inactive",
      "usage" :
      {

```

```

        "month" : 0,
        "today" : 0,
        "total" : 0,
        "week" : 0
    }
},
"peerAsHost" :
{
    "allowed" : true,
    "peerStatus" : "authenticated",
    "tunnelStatus" : "active",
    "usage" :
    {
        "month" : 0,
        "today" : 0,
        "total" : 0,
        "week" : 0
    }
},
"supportsHost" : true,
"uuid" : "F1C1290C-CA03-17B4-9989-A22222A28B74"
}
]

```

networksharing settings

Shows the current Pair & Share settings.

```

$ speedify_cli networksharing settings
{
    "alwaysOnDiscovery" : true,
    "autoPairBehavior" : "auto_user_team",
    "clientEnabled" : true,
    "displayName" : "peer device",
    "hostConnectCode" : "",
    "hostEnabled" : false,
    "pairRequestBehavior" : "accept"
}

```

networksharing startdiscovery

Begin discovering other devices on the local network.

```

$ speedify_cli networksharing startdiscovery
{
    "discoveryActive" : true
}

```

overflow <speed in mbps>

Speed in Mbps after which Secondary connections are not used.

```

$ speedify_cli overflow 10.0
{
    "allowChaChaEncryption" : true,
    "bondingMode" : "speed",
    "downstreamSubnets" :
    [
        {
            "address" : "192.168.8.8",
            "prefixLength" : 23
        }
    ]
}

```

```

    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
...

```

packetaggr <on|off>

The packetaggr command sets packet aggregation on/off.

```

$ speedify_cli packetaggr on
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
...

```

pep <on|off>

The pep command enables or disables the performance enhancing proxy (PEP).

```

$ speedify_cli pep on
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,

```

```

"forwardedPorts" :
[
  {
    "port" : 8001,
    "protocol" : "tcp"
  }
]
...

```

pep ips <add|rem|set> <ip address> ...

Configure IP addresses to use with the performance enhancing proxy (PEP).

```

$ speedify_cli pep ips add 8.8.8.8
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
}
...

```

pep ports <add|rem|set> [port[-portRangeEnd]/proto] ...

Configure ports to use with the performance enhancing proxy (PEP).

```

$ speedify_cli pep ports add 8080/tcp
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
}
...

```

pep dscp <add|rem|set> <dscp value 0-63>

Configure DSCP values (0-63) to use with the performance enhancing proxy (PEP).

```
$ speedify_cli pep dscp add 46
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
}
...
```

packetpool <small|default|large>

The `packetpool` command controls the size of the packet pool. A larger pool can provide better performance, but may use more memory.

```
$ speedify_cli packetpool default
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
}
...
```

ports [port/proto] ...

The `ports` command instructs Speedify to request public ports from a Dedicated (private) Speed Server. These settings only go into effect after a reconnect, and they are ignored by public Speed Servers. Requesting a port that is already taken by another user will lead to the connect request failing, and state will return to `LOGGED_IN`. Calling the `ports` command with no additional parameters will clear the port forward requests.

```
$ speedify_cli ports 8001/tcp
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
}
...
```

priorityoverflow <speed in mbps>

Speed in Mbps after which Secondary connections are not used for priority traffic.

```
$ speedify_cli priorityoverflow 10.0
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
}
...
```

privacy advancedIspStats <on|off>

Enables or disables advanced ISP statistics. When enabled, extra stats are available from some ISPs, such as Starlink.

```
$ speedify_cli privacy advancedIspStats on
{
  "advancedIspStats" : true,
  "apiProtection" : true,
  "appAnalytics" : false,
  "dnsAddresses" :
  [
  ]
}
```

```
[
  "8.8.8.8"
],
"requestToDisableDoH" : false,
"residentialProxyEnabled" : true
}
```

privacy apiProtection <on|off>

Enables or disables protection on the Speedify API port. When disabled, speedify_cli can be used to control a remote Speedify installation.

```
$ speedify_cli privacy apiProtection on
{
  "advancedIspStats" : true,
  "apiProtection" : true,
  "appAnalytics" : false,
  "dnsAddresses" :
  [
    "8.8.8.8"
  ],
  "requestToDisableDoH" : false,
  "residentialProxyEnabled" : true
}
```

privacy requestToDisableDoH <on|off>

If the Speedify VPN connection should request that browsers disable DNS over HTTPS. Enabling this can help streaming and streamingbypass rules function.

```
$ speedify_cli privacy requestToDisableDoH on
{
  "advancedIspStats" : true,
  "apiProtection" : true,
  "appAnalytics" : false,
  "dnsAddresses" :
  [
    "8.8.8.8"
  ],
  "requestToDisableDoH" : false,
  "residentialProxyEnabled" : true
}
```

refresh oauth [access token]

Accepts a new security token for the same user to be used when communicating with servers. If changing users, use login oauth instead.

```
$ speedify_cli refresh oauth abdef1234567890
```

refresh servers

The refresh servers command updates the list of available servers from the directory, bypassing any cached data.

```
$ speedify_cli refresh servers
{
  "private" :
  [
    {
      "city" : "nova",
      "country" : "us",
```

```

    "dataCenter" : "soladrive-nova",
    "friendlyName" : "United States - Northern Virginia #17",
    "isPremium" : false,
    "isPrivate" : true,
    "isSelfHosted" : false,
    "num" : 17,
    "tag" : "privateus-nova-17",
    "torrentAllowed" : true
  },
  {
    "city" : "Cedar_Knolls",
    "country" : "us",
    "dataCenter" : "Speedify",
    "friendlyName" : "United States - Cedar Knolls #1",
    ...

```

residentialproxy <on|off>

The residentialproxy command enables or disables the use of residential proxy if available on the speed server.

```

$ speedify_cli residentialproxy on
{
  "advancedIspStats" : true,
  "apiProtection" : true,
  "appAnalytics" : false,
  "dnsAddresses" :
  [
    "8.8.8.8"
  ],
  "requestToDisableDoH" : false,
  "residentialProxyEnabled" : true
}

```

residentialproxy ports <add|rem|set> [port[-portRangeEnd]/proto] ...

Configure ports to use with residential proxy.

```

$ speedify_cli residentialproxy ports add 8080/tcp
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
}
...

```

route default <on|off>

Configures whether Speedify will obtain a 'default' route to the Internet over the VPN adapter.

```
$ speedify_cli route default on
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
  ...
}
```

log erase

Deletes the Speedify daemon's log files.

```
$ speedify_cli log erase
```

log daemon <file size> <files per daemon> <total files> <verbose|info|warn|error>

The log daemon configures the size and number of the Speedify daemon's log files.

```
$ speedify_cli log daemon 3145728 7 9 info
{
  "daemon" :
  {
    "fileSize" : 3145728,
    "filesPerDaemon" : 7,
    "logLevel" : 1,
    "totalFiles" : 9
  }
}
```

show servers

The show servers command retrieves the current list of Speed Servers. If you have access to any Dedicated Speed Servers, they appear in a private array. The public pool of Speed Servers appear in a public array.

```
$ speedify_cli show servers
{
  "private" :
  [
    {
      "city" : "nova",
      "country" : "us",
      "dataCenter" : "soladrive-nova",
      "friendlyName" : "United States - Northern Virginia #17",

```

```

        "isPremium" : false,
        "isPrivate" : true,
        "isSelfHosted" : false,
        "num" : 17,
        "tag" : "privateus-nova-17",
        "torrentAllowed" : true
    },
    {
        "city" : "Cedar_Knolls",
        "country" : "us",
        "dataCenter" : "Speedify",
        "friendlyName" : "United States - Cedar Knolls #1",
        ...

```

show settings

The `show settings` command retrieves the current connection settings. These settings are sent to the server at connect time, and they can be retrieved at any time.

```

$ speedify_cli show settings
{
    "allowChaChaEncryption" : true,
    "bondingMode" : "speed",
    "downstreamSubnets" :
    [
        {
            "address" : "192.168.8.8",
            "prefixLength" : 23
        }
    ],
    "enableAutomaticPriority" : true,
    "enableDefaultRoute" : true,
    "encrypted" : false,
    "fixedDelay" : 100,
    "forwardedPorts" :
    [
        {
            "port" : 8001,
            "protocol" : "tcp"
        }
    ]
    ...

```

show privacy

Outputs privacy related settings

```

$ speedify_cli show privacy
{
    "advancedIspStats" : true,
    "apiProtection" : true,
    "appAnalytics" : false,
    "dnsAddresses" :
    [
        "8.8.8.8"
    ],
    "requestToDisableDoH" : false,
    "residentialProxyEnabled" : true
}

```

show adapters

The `show adapters` command allows the user to view all of the network adapters, and their settings and statistics.

```
$ speedify_cli show adapters
[
  {
    "adapterID" : "wlo1",
    "connectedNetworkBSSID" : "",
    "connectedNetworkName" : "",
    "dataUsage" :
    {
      "overlimitRatelimit" : 0,
      "usageDaily" : 202020,
      "usageDailyBoost" : 0,
      "usageDailyLimit" : 0,
      "usageMonthly" : 202020,
      "usageMonthlyLimit" : 0,
      "usageMonthlyResetDay" : 0
    },
    "description" : "wlo1",
    "directionalSettings" :
    {
      "download" : "on",
      "upload" : "on"
    }
  }
  ...
]
```

show currentserver

The `show currentserver` command displays the last server Speedify was connected (which, if you are connected is the current server).

```
$ speedify_cli show currentserver
{
  "city" : "nyc",
  "country" : "us",
  "dataCenter" : "edis-nyc",
  "dnsIP" :
  [
    "10.202.8.8"
  ],
  "friendlyName" : "United States - New York City #59",
  "isPremium" : false,
  "isPrivate" : false,
  "num" : 59,
  "publicIP" :
  [
    "151.236.8.8"
  ],
  "tag" : "us-nyc-59",
  "torrentAllowed" : false
}
```

show user

Outputs information about the currently logged in user.

```
$ speedify_cli show user
{
  "action" : "none",
  "bytesAvailable" : -1,
}
```

```
"bytesUsed" : 1624500292,
"dataPeriodEnd" : "2026-05-01",
"email" : "****@connectify.me",
"isAutoAccount" : false,
"isTeam" : true,
"minutesAvailable" : -1,
"minutesUsed" : 352412,
"paymentType" : "yearly",
"subscriptionEndDate" : "2026-04-13"
}
```

show directory

The `show directory` command shows the current directory server.

```
$ speedify_cli show directory
{
  "domain" : "",
  "enableEsni" : false,
  "gatewayUri" : ""
}
```

show connectmethod

The `show connectmethod` command displays the stored connectmethod (the default settings for `connect`).

```
$ speedify_cli show connectmethod
{
  "city" : "",
  "connectMethod" : "closest",
  "country" : "",
  "num" : 0
}
```

show streamingbypass

The `show streamingbypass` command displays current streaming bypass service settings.

```
$ speedify_cli show streamingbypass
{
  "domainWatchlistEnabled" : true,
  "domains" :
  [
    "hulu.com"
  ],
  "ipv4" :
  [
    "8.8.8.8",
    "170.170.8.8"
  ],
  "ipv6" : [],
  "ports" :
  [
    {
      "port" : 4800,
      "protocol" : "tcp"
    }
  ],
  "services" :
  ...
}
```

show disconnect

Displays the reason for the last disconnect.

```
$ speedify_cli show disconnect
{
  "disconnectReason" : "USERINITIATED"
}
```

show streaming

The show streaming command displays current streaming mode settings.

```
$ speedify_cli show streaming
{
  "domains" :
  [
    "mynewstreamingservice.com"
  ],
  "ipv4" :
  [
    "8.8.8.8"
  ],
  "ipv6" :
  [
    "2001:4860:4860::8888"
  ],
  "ports" :
  [
    {
      "port" : 8200,
      "protocol" : "tcp"
    }
  ]
  ...
}
```

show speedtest

The show speedtest command displays the last speed test results.

```
$ speedify_cli show speedtest
[
  {
    "city" : "newark",
    "country" : "us",
    "downloadSpeed" : 0.0,
    "fps" : 30,
    "isError" : false,
    "jitter" : 1,
    "latency" : 23,
    "loss" : 0.058882261078825424,
    "numConnections" : 1,
    "resolution" : "4K",
    "time" : 1775054668,
    "type" : "streaming",
    "uploadSpeed" : 24998865.0079444945
  },
  {
    "city" : "newark",
    "country" : "us",
    "downloadSpeed" : 59916517.568674102,
    ...
}
```

show logsettings

The `show dscp` command retrieves the current log settings.

```
$ speedify_cli show logsettings
{
  "daemon" :
  {
    "fileSize" : 3145728,
    "filesPerDaemon" : 7,
    "logLevel" : 1,
    "totalFiles" : 9
  }
}
```

show dscp

The `show dscp` command retrieves the current dscp settings.

```
$ speedify_cli show dscp
{
  "dscpQueues" :
  [
    {
      "dscp" : 0,
      "priority" : "on",
      "replication" : "auto",
      "retransmissionAttempts" : 2
    }
  ],
  "monitoringTags" :
  [
    46
  ]
}
```

show fixeddelay

Shows the current rules for including traffic in the jitter buffer. Rules for domains, ports, and IP addresses are supported.

```
$ speedify_cli show fixeddelay
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
  ...
}
```

show trafficrules

Shows the current traffic rules setting.

```
$ speedify_cli show trafficrules
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
  ...
}
```

show pep

The `show pep` command displays the current performance enhancing proxy (PEP) settings, including configured IPs, ports, and DSCP values.

```
$ speedify_cli show pep
{
  "trafficRules" :
  [
    {
      "actions" :
      [
        {
          "type" : "fixed_delay"
        }
      ],
      "conditions" :
      [
        {
          "type" : "ports",
          "value" :
          [
            {
              "port" : 9000,
              "portRangeEnd" : 9010,
              "proto" : "udp"
            }
          ]
        }
      ]
    }
  ]
  ...
}
```

show systemdata

The `show systemdata` command displays system data including the current device role.

```
$ speedify_cli show systemdata
{
```

```
"model": "Example Device",
"role" : "client",
"uuid": "00000000-0000-0000-0000-000000000000",
"vendor": "Example Vendor"
}
```

speedtest [adapter id]

Runs a speed test over the VPN tunnel, using a bundled iPerf3 client.

```
$ speedify_cli speedtest
[
  {
    "city" : "nyc",
    "country" : "us",
    "downloadSpeed" : 42022570.323507965,
    "isError" : false,
    "latency" : 24,
    "numConnections" : 1,
    "time" : 1775055550,
    "type" : "speed",
    "uploadSpeed" : 23210585.966454245
  }
]
```

streamtest [adapter id]

Runs a stream test over the VPN tunnel, using a bundled iPerf3 client. The streamtest is emulating broadcasting a live stream; it sends 60 Mbps of UDP traffic and measures the results.

```
$ speedify_cli streamtest
[
  {
    "city" : "nyc",
    "country" : "us",
    "downloadSpeed" : 0.0,
    "fps" : 60,
    "isError" : false,
    "jitter" : 2,
    "latency" : 23,
    "loss" : 0.0,
    "numConnections" : 1,
    "resolution" : "1080p",
    "time" : 1775055625,
    "type" : "streaming",
    "uploadSpeed" : 8999604.4033230096
  }
]
```

startupconnect <on|off>

The `startupconnect` option tells Speedify if it should connect automatically at startup or not. It prints the current settings immediately after the request is made.

```
$ speedify_cli startupconnect on
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
```

```

        "address" : "192.168.8.8",
        "prefixLength" : 23
    }
],
"enableAutomaticPriority" : true,
"enableDefaultRoute" : true,
"encrypted" : false,
"fixedDelay" : 100,
"forwardedPorts" :
[
    {
        "port" : 8001,
        "protocol" : "tcp"
    }
]
...

```

state

The `state` command retrieves the current state of the connection. Possible states are `LOGGED_OUT`, `LOGGING_IN`, `LOGGED_IN`, `AUTO_CONNECTING`, `CONNECTING`, `DISCONNECTING`, `CONNECTED`, `OVERLIMIT`, and `UNKNOWN`

```

$ speedify_cli state
{
    "state" : "CONNECTED"
}

```

`stats [historic [max-seconds] [<wantedGuid> ...]] | [[durationSeconds] [networksharing] [current|day|week|month|total|<periodHours>] ... [<tunnel-guid> ...]] | [streaming] | [session [current|day|week|month|total|<periodHours>]]`

The `stats` command subscribes to a feed of connection and session statistics. By default, this feed will continue until the `speedify_cli` process is terminated, but an optional parameter can be given to stop and exit after the given number of seconds. This can be useful to monitor how many connections are being utilized by Speedify, and what their current network activity level is in bytes per second. You can specify up to 5 time periods to receive stats over.

```

$ speedify_cli stats 1
[
    "state",
    {
        "state" : "CONNECTED"
    }
]

[
    "connection_stats",
    {
        "connections" :
        [
            {
                "adapterID" : "wlo1",
                "bondingFiltered" : false,
                "congested" : false,
                "connected" : true,
                "connectionID" : "wlo1%192.168.1.0/24",
                "downloadCongested" : false,
                "inFlight" : 37440,
                ...
            }
        ]
    }
]

```

streaming domains <add|rem|set> <domain> ...

Configure extra domains to be treated as high priority streams when in streaming mode.

```
$ speedify_cli streaming domains add mynewstreamingservice.com
{
  "domains" :
  [
    "mynewstreamingservice.com"
  ],
  "ipv4" :
  [
    "8.8.8.8"
  ],
  "ipv6" :
  [
    "2001:4860:4860::8888"
  ],
  "ports" :
  [
    {
      "port" : 8200,
      "protocol" : "tcp"
    }
  ]
}
...
```

streaming ipv4 <add|rem|set> <ip address> ...

Configure extra IPv4 addresses to be treated as high priority streams when in streaming mode.

```
$ speedify_cli streaming ipv4 add 8.8.8.8
{
  "domains" :
  [
    "mynewstreamingservice.com"
  ],
  "ipv4" :
  [
    "8.8.8.8"
  ],
  "ipv6" :
  [
    "2001:4860:4860::8888"
  ],
  "ports" :
  [
    {
      "port" : 8200,
      "protocol" : "tcp"
    }
  ]
}
...
```

streaming ipv6 <add|rem|set> <ip address> ...

Configure extra IPv6 addresses to be treated as high priority streams when in streaming mode.

```
$ speedify_cli streaming ipv6 add 2001:4860:4860::8888
{
  "domains" :
```

```

[
  "mynewstreamingservice.com"
],
"ipv4" :
[
  "8.8.8.8"
],
"ipv6" :
[
  "2001:4860:4860::8888"
],
"ports" :
[
  {
    "port" : 8200,
    "protocol" : "tcp"
  }
]
...

```

streaming ports <add|rem|set> [port[-portRangeEnd]/proto] ...

Configure extra ports to be treated as high priority streams when in streaming mode.

```
$ speedify_cli streaming ports add 8200/tcp
```

```

{
  "domains" :
  [
    "mynewstreamingservice.com"
  ],
  "ipv4" :
  [
    "8.8.8.8"
  ],
  "ipv6" :
  [
    "2001:4860:4860::8888"
  ],
  "ports" :
  [
    {
      "port" : 8200,
      "protocol" : "tcp"
    }
  ]
}
...

```

streamingbypass domains <add|rem|set> [<domain> ...]

Configure extra domains to bypass the VPN.

```
$ speedify_cli streamingbypass domains add hulu.com
```

```

{
  "domainWatchlistEnabled" : true,
  "domains" :
  [
    "hulu.com"
  ],
  "ipv4" :
  [

```

```

    "8.8.8.8",
    "170.170.8.8"
  ],
  "ipv6" : [],
  "ports" :
  [
    {
      "port" : 4800,
      "protocol" : "tcp"
    }
  ],
  "services" :
  ...

```

streamingbypass ipv4 <add|rem|set> <ip address> ...

Configure extra IPv4 addresses to bypass the VPN.

```

$ speedify_cli streamingbypass ipv4 add 8.8.8.8
{
  "domainWatchlistEnabled" : true,
  "domains" :
  [
    "hulu.com"
  ],
  "ipv4" :
  [
    "8.8.8.8",
    "170.170.8.8"
  ],
  "ipv6" : [],
  "ports" :
  [
    {
      "port" : 4800,
      "protocol" : "tcp"
    }
  ],
  "services" :
  ...

```

streamingbypass ipv6 <add|rem|set> <ip address> ...

Configure extra IPv6 addresses to bypass the VPN.

```

$ speedify_cli streamingbypass ipv4 add 2001:4860:4860::8888
{
  "domainWatchlistEnabled" : true,
  "domains" :
  [
    "hulu.com"
  ],
  "ipv4" :
  [
    "8.8.8.8",
    "170.170.8.8"
  ],
  "ipv6" : [],
  "ports" :
  [

```

```

    {
      "port" : 4800,
      "protocol" : "tcp"
    }
  ],
  "services" :
...

```

streamingbypass ports <add|rem|set> <port[-portRangeEnd]/proto> ...

Configure extra ports to bypass the VPN.

```

$ speedify_cli streamingbypass ports add 4800/tcp
{
  "domainWatchlistEnabled" : true,
  "domains" :
  [
    "hulu.com"
  ],
  "ipv4" :
  [
    "8.8.8.8",
    "170.170.8.8"
  ],
  "ipv6" : [],
  "ports" :
  [
    {
      "port" : 4800,
      "protocol" : "tcp"
    }
  ],
  "services" :
...

```

streamingbypass service <enable|disable|service name> [<on|off>]

Configures whether Speedify will allow traffic from a service to bypass the VPN.

```

$ speedify_cli streamingbypass service Netflix on
{
  "domainWatchlistEnabled" : true,
  "domains" :
  [
    "hulu.com"
  ],
  "ipv4" :
  [
    "8.8.8.8",
    "170.170.8.8"
  ],
  "ipv6" : [],
  "ports" :
  [
    {
      "port" : 4800,
      "protocol" : "tcp"
    }
  ],
  "services" :
...

```

subnets [subnet/length] ...

Configures a group of subnets connected to this machine as accessible by other clients on a private server. Only for advanced enterprise routing scenarios.

```
$ speedify_cli subnets 192.168.202.1/23
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
}
...
```

targetconnections <number upload connections> <number download connections>

The `targetconnections` command controls the amount of connections Speedify will attempt to use for upload and download, respectively.

```
$ speedify_cli targetconnections 5 5
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
}
...
```

transport <auto|tcp|tcp-multi|udp|https>

The `transport` command instructs Speedify to choose between one of the network protocols `auto`, `tcp`, `tcp-multi`, `udp`, or `https`. The `transport_mode` value is set accordingly based on the user's selection.

```

$ speedify_cli transport udp
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
...

```

transportretry <time in seconds>

The `transportretry` command sets the backoff timeout limit in seconds when Speedify fails to establish VPN transport connections over an Internet connection. With each subsequent failure, the wait time increases before attempting to establish a transport over that connection again. This call sets the upper limit on that backoff timeout.

```

$ speedify_cli transportretry 300
{
  "allowChaChaEncryption" : true,
  "bondingMode" : "speed",
  "downstreamSubnets" :
  [
    {
      "address" : "192.168.8.8",
      "prefixLength" : 23
    }
  ],
  "enableAutomaticPriority" : true,
  "enableDefaultRoute" : true,
  "encrypted" : false,
  "fixedDelay" : 100,
  "forwardedPorts" :
  [
    {
      "port" : 8001,
      "protocol" : "tcp"
    }
  ]
...

```

version

The `version` command can be used to verify the version of Speedify that is installed and running.

```

$ speedify_cli version
{
  "bug" : 0,
  "build" : 154,

```

```
"maj" : 16,  
"min" : 5  
}
```

Copyright Connectify, Inc.